

Search Suggestions System [\(View\)](#)

You are given an array of strings `products` and a string `searchWord`.

Design a system that suggests at most three product names from `products` after each character of `searchWord` is typed. Suggested products should have common prefix with `searchWord`. If there are more than three products with a common prefix return the three lexicographically minimums products.

Return *a list of lists of the suggested products after each character of `searchWord` is typed*.

Example 1:

Input: `products = ["mobile","mouse","moneypot","monitor","mousepad"], searchWord = "mouse"`

Output: `[
 ["mobile","moneypot","monitor"],
 ["mobile","moneypot","monitor"],
 ["mouse","mousepad"],
 ["mouse","mousepad"],
 ["mouse","mousepad"]
]`

Explanation: products sorted lexicographically = `["mobile","moneypot","monitor","mouse","mousepad"]`

After typing m and mo all products match and we show user `["mobile","moneypot","monitor"]`

After typing mou, mous and mouse the system suggests `["mouse","mousepad"]`

Example 2:

Input: `products = ["havana"], searchWord = "havana"`

Output: `[["havana"],["havana"],["havana"],["havana"],["havana"],["havana"]]`

Example 3:

Input: `products = ["bags", "baggage", "banner", "box", "cloths"], searchWord = "bags"`

Output:

```
[["baggage", "bags", "banner"], ["baggage", "bags", "banner"], ["baggage", "bags"], ["bags"]]
```

Constraints:

- `1 <= products.length <= 1000`
- `1 <= products[i].length <= 3000`
- `1 <= sum(products[i].length) <= 2 * 104`
- All the strings of `products` are **unique**.
- `products[i]` consists of lowercase English letters.
- `1 <= searchWord.length <= 1000`
- `searchWord` consists of lowercase English letters.